6LoRa: Full Stack IPv6 Networking with DSME-LoRa on Low Power IoT Nodes

José Álamos
HAW Hamburg
jose.alamos@haw-hamburg.de

Thomas C. Schmidt
HAW Hamburg
t.schmidt@haw-hamburg.de

Matthias Wählisch
TU Dresden & Barkhausen Institut
m.waehlisch@tu-dresden.de

Abstract

Long range wireless transmission techniques such as LoRa are preferential candidates for a substantial class of IoT applications, as they avoid the complexity of multi-hop wireless forwarding. The existing network solutions for LoRa, however, are not suitable for peer-to-peer communication, which is a key requirement for many IoT applications. In this work, we propose a networking system – 6LoRa, that enables IPv6 communication over LoRa. We present a full stack system implementation on RIOT OS and evaluate the system on a real testbed using realistic application scenarios with CoAP. Our findings confirm that our approach outperforms existing solutions in terms of transmission delay and packet reception ratio at comparable energy consumption.

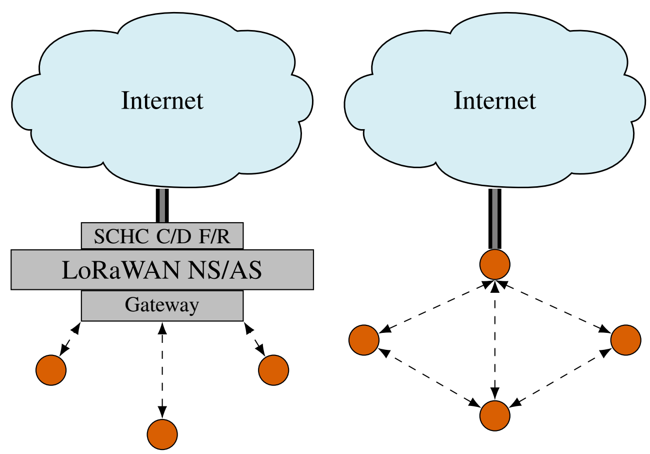
Categories and Subject Descriptors

C.2 [**Computer-Communication Networks**]: Network Architecture and Design; C.2 [**Computer-Communication Networks**]: Network Protocols; D.2 [**Software Engineering**]: Interoperability

General Terms

Design, Measurement, Reliability

Keywords. Wireless, LPWAN, MAC layer, network experimentation



(a) SCHC-LoRaWAN (b) 6LoRa

Figure 1. Network architecture of SCHC-LoRaWAN (left) and our solution (right).

1 Introduction

Proliferation of proprietary IoT devices has led a fragmentation of IoT communication protocols, which challenges interoperation and slows down innovation. To address this problem, the Internet Engineering Task Force (IETF) has proposed adaptation layers to run IPv6 on top of a variety of IoT technologies such as IEEE 802.15.4 (6LoWPAN [15]) and LPWAN (SCHC [6]).

Beyond interoperation, IPv6 has two additional advantages. (*i*) It utilizes existing infrastructure, which is well-proven with many successful deployments; (*ii*) it enables the utilization of state of the art IoT application protocols such as the Constrained Application Protocol (CoAP) [18].

The LoRaWAN [11] LPWAN specification defines a cloud-based Media Access Control layer protocol on top of the wireless modulation LoRa, which is known to achieve long range transmissions range (km) at very low power (mJ). LoRaWAN has attracted special attention in the industry and academia for its open specification and low hardware cost.

While SCHC successfully enables transmission of IPv6 packets over LoRaWAN, its centralized architecture (Figure 1, left) challenges applications that rely on downlink communication such as automation control scenarios. The architecture does not allow for direct peer-to-peer communication between IoT devices, which is a common pattern in ad hoc networks. In consequence, the network must forward messages from a transmitting device to the cloud and then back to the target device. This approach exhibits three problems. (*i*) Downlink communication consumes base stations transmission budget and prevents concurrent reception of uplink frames; (*ii*) it hinders the deployment of edge devices and leads to increased costs in cloud infrastructure; (*iii*) in many remote areas, cellular networks are the only feasible uplink option for the base stations, which renders communication between IoT devices useless on poor Internet connectivity.

Recently, we proposed DSME-LoRa [1], an adaptation layer to operate IEEE 802.15.4e Deterministic and Synchronous Multichannel Extension (DSME) [10] on top of the LoRa modulation. This obsoletes the LoRaWAN network emulation. In this paper, we go one step further and develop 6LoRa, a system to enable IPv6 communication over DSME-LoRa networks. (Figure 1, right).

The contributions of this paper are:

1. A system architecture for transmitting IPv6 frames over DSME-LoRa (6LoRa).
2. An implementation of 6LoRa in the RIOT operating system.
3. A comparative evaluation on real hardware of the system against SCHC-LoRaWAN.

The remainder of this paper is structured as follows. We recap the background with reviewing related work on LoRaWAN, IEEE 802.15.4 DSME and CoAP/IPv6 in Section 2. Section 3 introduces the system design. Section 4 describes the implementation in RIOT. We show the results of our evaluation on real hardware in Section 5 and conclude with an outlook in Section 6.

2 Background and related work

LoRaWAN. LoRaWAN contributes a networking substrate to the long range transmission layer of LoRa [11]. Devices and cloud applications exchange data through a backhaul infrastructure consisting of gateways (base stations), a network server (network controller), and an application server (cloud services). Although communication is bidirectional, the architecture favors uplink communication.

LoRaWAN offers three communication classes, namely A, B and C, that trade-off downlink latency and energy consumption. Class A seizes uplink packets for polling downlink messages. Following packet transmission, devices open two reception windows (RX1 and RX2) to receive a downlink packet. If the network server has multiple packets to deliver, it sets the frame pending bit in the next downlink, which prompts the receiving device to schedule an uplink as soon as possible. Class B devices open reception windows at regular intervals. Class C devices keep the reception window open all the time.

Mikhaylov *et al.* [12] shows that downlink traffic compromises performance of uplinks, which hinders deterministic communication. Although LoRaWAN class B decreases downlink latencies compared to class A [4], it suffers from scalability issues [5, 19]. Vincenzo *et al.* [21] propose countermeasures for the limitations of LoRaWAN downlink traffic by adding multiple gateways and providing a gateway selection mechanism. This increases deployment costs. Zorbas *et al.* [22] present TS-LoRa, which adds a time-slotted mechanism to LoRaWAN and achieves more than 99% packet delivery ratio. Still, it does not address the downlink limitations of LoRaWAN.

IEEE 802.15.4 DSME. The IEEE 802.15.4e Deterministic Synchronous Multichannel Extension (DSME) [10] defines a beacon-synchronized link layer. The MAC initiates a superframe structure, which reserves time for contention-based (CSMA/CA) and contention-free (GTS) transmission. The superframe structure repeats indefinitely.

CSMA/CA transmissions are carried out in a common channel, whereas GTS transmissions are carried out in dedicated time-frequency slots. Devices must negotiate a slot with the receiver prior GTS transmissions. The DSME MAC allows for grouping superframes into a multisuperframe, which extends the number of available GTS at the cost of increased transmission latency.

DSME devices may operate in two modes, namely coordinator and child. Coordinators are responsible for network formation, maintenance and data routing between children and devices beyond the radio range of each other. The DSME MAC supports star, mesh and cluster-tree topologies depending on the arrangement of coordinators and children. Among coordinators, one assumes a significant role known as the PAN Coordinator, which dictates the superframe structure for the entire network.

Battaglia *et al.* [3] present two novel extensions for the DSME MAC, which address scalability issues in GTS handling in networks with a high number of nodes and periodic flows. Vallati *et al.* [20] discover inefficiencies during DSME network formation and propose effective mitigations. Alamos *et al.* [1] present and evaluate DSME-LoRa, which defines an adaptation layer to interface the DSME MAC with a standard LoRa transceiver. We employ the adaptation layer as a basis for our work.

IoT-centric IPv6 communication. SCHC (RFC 8724 [14]) describes a generic header compression and fragmentation of IPv6 packets for transmission over a generic LPWAN. It employs a set of out of band static rules (context) to instruct devices on the procedure to compress and decompress IPv6 packets, selecting the most suitable rule. RFC9011 [6] describes the LoRaWAN adaptation for SCHC. The 6LoWPAN protocol [9] enables the transmission of IPv6 packets over IEEE 802.15.4 networks by using a set of headers (*i.e.,* IP Header Compression, Next Header Compression for UDP) to facilitate compression and decompression, fragmentation and reassembly of transmitted packets. The protocol supports both stateful and stateless compression schemes. In contrast to 6LoWPAN, the compression scheme of SCHC performs better [8] and supports compression of the CoAP header [13]. This has motivated research efforts towards the transmission of SCHC-compressed packets over IEEE 802.15.4 networks [7].

The Constrained Application Protocol [18] (CoAP) is a lightweight Web protocol based on UDP. The protocol follows a Request/Response model and incorporates REST-like methods, including GET, POST and PUT.

3 System design

Figure 2 summarizes the system architecture for a 6LoRa network stack. The design caters for IPv6 packet transmission over LoRa.

We utilize the DSME MAC to enable reliable communication of IPv6 packets in LoRa frames. The MAC interacts with the LoRa transceiver via the DSME-LoRa adaptation

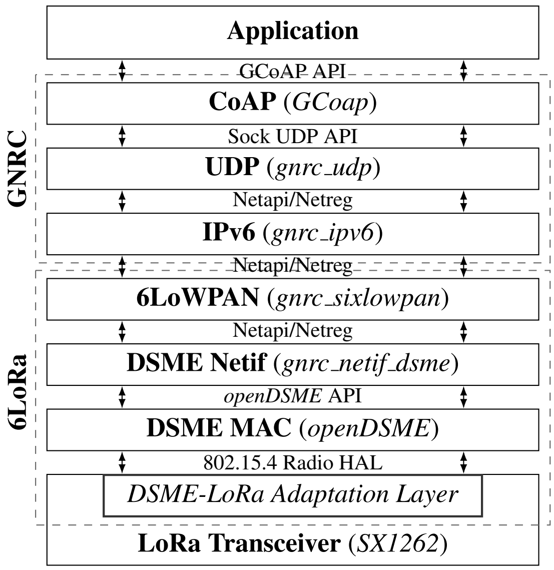
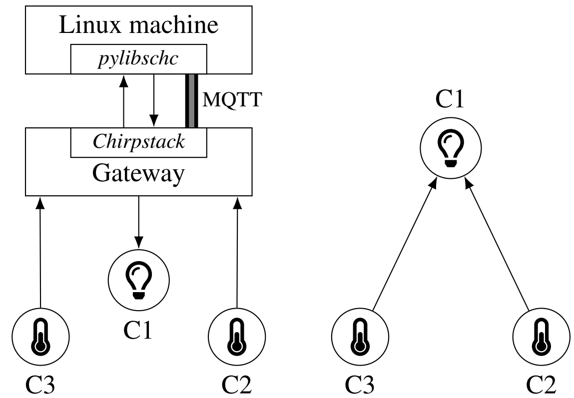


Figure 2. System architecture of a 6LoRa network stack. The figure includes software components and interfaces of the implementation in the RIOT operating system.



(a) SCHC-LoRaWAN (b) 6LoRa

Figure 3. Network topology and traffic flow of the SCHC-LoRaWAN network (left) and 6LoRa network (right) used in our evaluation. C1 corresponds to the cluster with three actuators (💡), while C2 and C3 represent clusters with eight and four sensors (🌡), respectively.

layer [1]. The DSME network interface grants full control to the upper layer of the transmission pattern (CSMA/CA or slotted), the superframe duration and slot allocation. To enable compression and fragmentation of IPv6 packets, we use the 6LoWPAN [9] adaptation layer. This is required to reduce link stress and decrease time on air of LoRa frames. On top of 6LoRa, we consider a standard IoT network stack consisting of CoAP via UDP and IPv6. Similar to traditional 6LoWPAN networks, devices in the network may act as routers to forward traffic between devices or to the Internet (not evaluated in this work).

4 Implementation

We extend the implementation of DSME-LoRa in the RIOT operating system [2] to enable IPv6 support via the network stack (GNRC), as shown in Figure 2. The implementation provides DSME support through the *openDSME* package and implements the *802.15.4 Radio HAL* interface with the DSME-LoRa adaptation layer, in order to interface the DSME MAC with the LoRa transceiver (*SX1262*). The GNRC network stack runs each protocol layer on a dedicated thread and uses inter process communication via a common API (GNRC Netapi/Netreg) to communicate between layers. This approach facilitates protocol implementation.

We add IPv6 support to the existing DSME-LoRa network interface (*gnrc_netif_dsme*). For that, we take care of three aspects. (*i*) the initialization of the necessary IPv6 Information Base, such as hop limit and network address; (*ii*) the configuration of the interface for use with 6LoWPAN and (*iii*) perform broadcast transmission for packets which where already tagged by the network stack as multicast.

The 6LoRa network stack utilizes the UDP (`gnrc_udp`) and IPv6 (`gnrc_ipv6`) components of the network stack for the transmission of CoAP frames and 6LoWPAN (`gnrc_sixlowpan`) to transmit compressed IPv6 packets over the DSME MAC.

GNRC implements the Sock UDP API (`sock_udp`), which exposes network stack agnostic UDP sockets. We utilize the RIOT CoAP implementation (GCoAP), which uses the Sock UDP API to send and receive CoAP packets.

5 Evaluation

We assess the performance of CoAP communication over SCHC-LoRaWAN and 6LoRa devices. We also compare memory requirements and power consumption of both stacks. In this study, fragmentation is not evaluated.

5.1 Experimental setup

Experiments are conducted using commercial off-the-shelf platforms (Nucleo-WL55JC1). The platform is equipped with a dual core ARM Cortex-M4F and ARM Cortex-M0+ CPU, with a clock speed of 48 MHz and a memory capacity of 256 kB of ROM/64 kB of RAM. Due to the absence of dual-core support in RIOT, we solely utilize the Cortex-M4F core. The platform is further equipped with a LoRa transceiver (SX1262).

Fifteen devices were deployed in the university facilities, grouped in three clusters. The first cluster (C1) consists of three LoRa devices that function as actuator devices. The second cluster (C2) includes eight LoRa devices and is located on the floor above C1, approximately 60 m away of C1. The third cluster (C3) is composed of four LoRa devices and is deployed without line-of-sight on the same floor of C1, approximately 12 m away. Devices in C2 an C3 operate as sensor devices.

Each sensor device from C2 and C3 sends 12 bytes in non-confirmed CoAP POST packets to an actuator from C1. Out of the twelve sensors, six of them send data exclusively to the first actuator. Among the remaining sensors, three of them send data to the second actuator, while the other three send data to the third actuator. To minimize wireless traffic, we configure CoAP to elide responses. The transmission intervals are governed by a uniform distribution characterized by means of 10 s and 20 s, and a range of plus or minus 2 seconds. The LoRaWAN infrastructure forwards traffic between end devices and the SCHC endpoint (Figure 3(a)), while 6LoRa devices employ direct peer-to-peer communication (Figure 3(b)).

We deploy a RAK2245 LoRaWAN gateway next to the C1 cluster, which runs the *Chirpstack* LoRaWAN network server (v3.15.1) and application server (v3.17.2). The network server is configured to use all eight LoRaWAN channels that the gateway supports. In order to minimize the time on air of downlink packets, the datarate of the second reception window (RX2) channel is set to 5 (SF7BW125). The evaluation does not consider class B due to limited support in RIOT.

We employ the *libschc* library [16] for the SCHC compression and decompression in the constrained devices. Additionally, we utilize a simple Python implementation that uses *pylibschc*, a *libschc* wrapper, to communicate with Chirpstack via MQTT and expose a SCHC endpoint on a Linux machine.

We configure a single SCHC rule for the compression of IPv6 and UDP headers, that achieves the maximum level of compression attainable by SCHC for our experiment. Due to its incomplete CoAP support, *libschc* does not compress the CoAP header.

SCHC-LoRaWAN actuator devices operating under class A protocol employ polling packets with empty payload at a regular 10-second interval, in order to retrieve downlink packets from the network server.

In 6LoRa scenarios we evaluate only GTS transmissions. The choice is made to mitigate the detrimental effects of concurrent channel access in CSMA/CA, which degrades packet reception [1]. We set the superframe duration to the minimum feasible time (3.84 s) that allows transmissions of the CoAP POST packet within a GTS. Devices allocate only one GTS, which repeats at the superframe duration (3.84 s).

SCHC-LoRaWAN and 6LoRa devices do not request acknowledgements and transmit using the same PHY configuration (SF7BW125). For 6LoWPAN we use stateful, context-based compression for the IPv6 source and destination addresses.

5.2 Memory requirements

Figure 4 shows the memory requirements, separated into ROM (*text* + *data* segment) and RAM (*data* + *bss* segment), for SCHC-LoRaWAN and 6LoRa in our target platform (Nucleo-WL55JC1).

We evaluate only the memory requirements for the compression/decompression fragmentation/reassembly layer (SCHC, 6LoWPAN) and link layer (LoRaWAN, DSME-LoRa). Hence, we do not consider the memory requirements for components of the operating system (*i.e.,* GNRC network stack, RIOT core and LoRa driver), as these components do not vary significantly between the two firmwares.

The SCHC layer requires 13.62 kB of ROM and 3.60 kB of RAM. This layer includes the *libschc* library, which stores internal buffers in RAM, and the RIOT contrib code, which allocates ≈ 1 kB in RAM for the *libschc* thread stack.

The LoRaWAN stack (GNRC LoRaWAN) requires 10.61 kB of ROM. The stack operates frugal and requires

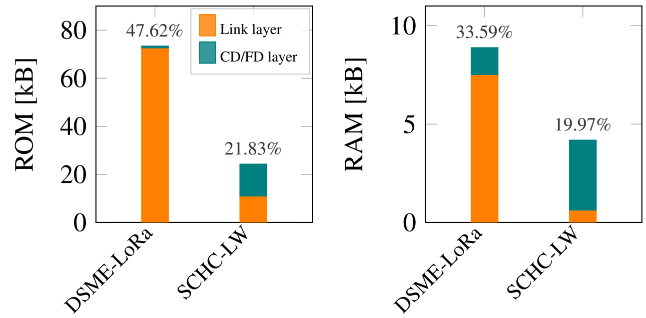


Figure 4. ROM (left) and RAM (right) requirements for SCHC-LoRaWAN and 6LoRa in RIOT, separated in compression/decompresion fragmentation/reassembly (CD/FR) and link layer, measured on a Nucleo-WL55JC1 platform (ARM Cortex-M4F). Relative values relate to the total firmware size.

only 0.58 kB of RAM to store the MAC descriptor and the MAC Information Base.

To summarize, the SCHC-LoRaWAN implementation utilizes a total of 24.23 kB (21.83%) of ROM and 4.18 kB (19,97%) of RAM.

The 6LoWPAN implementation (GNRC 6LoWPAN) requires 1.12 kB of ROM for the protocol logic and 1.41 kB of RAM for the thread stack.

The DSME-LoRa implementation includes *openDSME*, the contrib code for the platform abstraction and the DSME-LoRa Adaptation Layer. The implementation requires 72.19 kB of ROM and 7.47 kB of RAM, primarily attributed to the DSME MAC implementation. This MAC is more intricate compared to the LoRaWAN MAC and necessitates additional resources for storing internal data structures. To summarize, 6LoRa requires 73.31 kB (47.62%) of ROM and 8.88 kB (33.59%) of RAM.

5.3 Communication performance

In both scenarios, class C demonstrates a greater packet reception ratio compared to class A. Specifically, class C achieves an 85% ratio in the relaxed scenario (Figure 5, top center) and an 80% ratio in the stressed scenario (Figure 5,bottom center), while class A achieves a 77% ratio in the relaxed scenario (Figure 5,top left) and a 73% ratio in the stressed scenario (Figure 5, bottom left). The better performance of class C occurs for two reasons: (*i*) polling packets from class A actuators increases on-air traffic and consequently degrades channel quality, ultimately reducing data extraction ratio, and (*ii*) a fraction of class A downlink packets are scheduled in the first reception window (RX1), which utilizes the same frequency and datarate as uplink transmissions. As a result, uplink and downlink frames collide [17]. This does not occur for class C devices, as they receive most downlink packets in the same frequency of the second reception window (RX2), which uses a frequency that does not interfere with uplink transmissions.

A shorter transmission interval (10 s) decreases data extraction ratio (*i.e.,* successfully received uplink packets) for three reasons: (*i*) the increased on-air traffic from sensors increases uplink collisions; (*ii*) higher downlink traffic in-

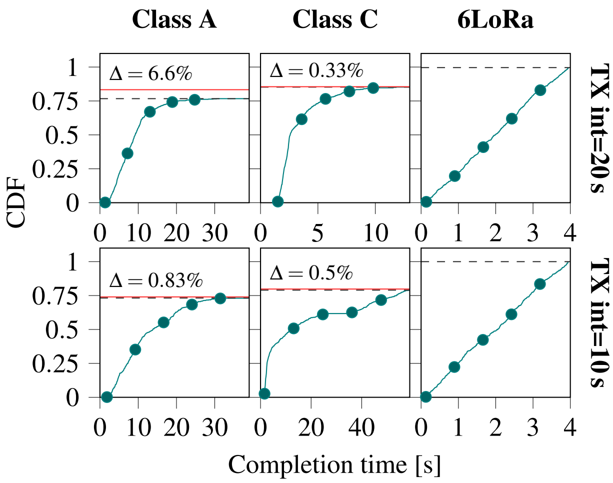


Figure 5. Distribution of completion time for transmitted CoAP packets over SCHC-LoRaWAN (class A and C) and 6LoRa, for varying transmission interval (20 s and 10 s). The intersection between curve and right axis relates to the packet reception ratio (dashed line). Data extraction ratio (*i.e.,* successful uplink packets) is shown for LoRaWAN (red line). Difference between red and dashed lines (Δ) relates to downlink losses.

creases transmission duty cycle at the gateway, which reduces its availability to receive uplink traffic; and (*iii*) the network server frequently sets the frame pending bit for class A devices, leading to increased polling packet traffic and degraded uplink reception. Observe that class A devices (Figure 5, left) experience higher downlink losses in the relaxed scenario (6.6%) than in the stressed scenario (0.83%). This outcome, which may appear counterintuitive, is attributed to the preference of the network server for the first reception window (RX1) for downlink traffic, which results in collisions with uplink traffic. In the stressed scenario, the network server schedules downlink traffic more often in the second reception window (RX2), whose traffic does not collide with uplink traffic.

The completion time is given by the sum of uplink transmission time (≈ 260 ms, measured from the beginning of the transmission until the reception on the SCHC endpoint), the SCHC endpoint processing time (≈ 5 ms), the downlink schedule delay (variable) and the downlink transmission time (≈ 1.3 s, which includes the gateway delay and the transmission time on air of the downlink packet).

The downlink schedule delay, which determines the completion time, depends on the downlink throughput and the downlink queue stress. Class A throttles downlink rate to the polling rate of the receiving device, whereas class C throttles the downlink rate to one downlink message every ≈ 2 s. As a result, even under low queue stress (*i.e.,* relaxed scenario, Figure 5, top), class C exhibits lower completion time (less than 7 s for 95% of packets), than class A (less than ≈ 17 s for 95% of packets).

In the stressed scenario, the downlink queue experiences high stress due to increased sensor traffic. For class A actuators, this influx of traffic has an impact on the delivery of polling packets. These combined effects lead to extended waiting times for packets in the downlink queue, ultimately resulting in a significant increase in completion time. The knee point at approximately 35 s for class C devices in the stressed scenario (Figure 5, bottom center) can be attributed to a continuous build up of downlink packets for the actuator with six sensor devices, which schedules in average at a shorter interval (≈ 1.67) than the throttling interval (2 s).

The completion time of 6LoRa transmissions is primarily determined by the transmission delay of the DSME MAC layer, which is influenced by the GTS schedule, and the time on air of the packet (≈ 118 ms). When the application schedules a transmission just before the occurrence of the next slot, the lowest completion time is achieved, resulting in less than 140 ms (Figure 5, right). Conversely, if the application schedules the transmission immediately after the occurrence of the next slot, the packet is held in the MAC queue until the next slot occurs (3.84 s after the last slot), resulting in a worst-case completion time of 100% of packets transmitted in less than ≈ 3.9 s. The completion time remains constant in both the relaxed and stressed scenarios because the inter-packet gap is always greater than the superframe duration. Consequently, all packets are scheduled when the MAC queue is empty, which effectively bounds the completion time to a value that aligns with the superframe duration.

In contrast to LoRaWAN traffic, DSME-LoRa traffic is not vulnerable to collisions thanks to the time-division multiple access scheme (TDMA) of GTS, which enables collision-free transmissions. The empirical results indicate a significant packet reception rate, reaching 99.7% for the relaxed scenario and 100% for the stressed scenario.

Table 1. Power consumption of SCHC-LoRaWAN (class A and C) , and 6LoRa devices, for sensor devices with transmission intervals (TXi) of 20 s and 10 s, and for actuator devices.

Device	TXi [s]	Power [mW]		
		SCHC-LoRaWAN		6LoRa
		Class A	*Class C*	
Sensor	20	0.49	12.87	1.33
Sensor	10	0.87	13.3	2.04
Actuator	-	0.54	12.41	2.93

5.4 Energy consumption

We utilize a digital multimeter (DMM7510 7 1/2) to measure the current at a sampling rate of 100 kHz in order to estimate the power consumption of the devices. Our analysis focused on the power consumption of sensors as well as actuators that receive data from three sensors each. To conserve energy, the 6LoRa devices in our analysis disable the transceiver during the contention access period (reserved for CSMA/CA transmissions).

Table 1 illustrates the power consumption (over base consumption) for SCHC-LoRaWAN and 6LoRa devices.

Unlike the DSME MAC, LoRaWAN class A and class C devices have a low maintenance overhead since the MAC does not require maintaining a continuous connection with the network server. Additionally, class A devices keep the transceiver off most of the time, resulting in the lowest maintenance overhead and consequently, the lowest power consumption across all scenarios. Although 6LoRa devices yield low transceiver duty cycle, the DSME MAC consumes energy in order to maintain the superframe structure (*i.e.,* synchronize to beacons and manage GTS). This results in a higher maintenance overhead for 6LoRa devices, which yields a higher power consumption than class A devices in all scenarios but does not prevent operation on batteries. Class C devices show the highest power consumption, as a result of the transceiver listening continuously.

The power consumption of SCHC-LoRaWAN sensors increases by ≈ 0.4 mW for the stressed scenario (10 s), whereas the power consumption of 6LoRa sensors increases by ≈ 0.7 mW. This is attributed to (*i*) the shorter time on air (108 ms) of the LoRaWAN packet as a result of the superior compression scheme of SCHC and (*ii*) inefficient handling in *openDSME* of the GTS schedule, where the transceiver is turned on too early. For this reason, the energy consumption per transmitted packet is lower for SCHC-LoRaWAN devices than for 6LoRa devices.

Receiving class A devices transmit polling packets (10 s transmission interval) which consumes ≈ 0.54 mW. Unlike data packets, polling packets do not carry a payload, resulting in a lower time on air and lower power consumption compared to class A sensor devices with 10 s transmission interval (≈ 0.87 mW). Class C devices consume ≈ 12.41 mW as a result of the continuous listening of the transceiver. Receiving 6LoRa devices consume approximately 4 times less energy (2.93 mW) than class C devices (12.41 mW) due to the transceiver being active approximately 25% of the time (one slot for beacon reception and three slots for GTS reception).

In conclusion, our findings validate that class A devices demonstrate the lowest power consumption, while class C devices exhibit the highest power consumption. Although 6LoRa is viable for battery-powered devices, the power consumption is more than 3 times higher than that of class A devices.

6 Conclusions and outlook

In this work, we proposed a system architecture (6LoRa) that enables IPv6 communication for the DSME MAC over LoRa. This overcomes the limitations of LoRaWAN, a widely used LPWAN technology. The evaluation on common off-the-shelf devices confirmed that our system renders better communication performance for peer-to-peer scenarios than SCHC-LoRaWAN. Our solution requires more memory resources than SCHC-LoRaWAN, although it remains compatible with constrained-devices. The proposed approach facilitates energy-efficient IPv6 communication over LoRa, obviating the need for additional backhaul infrastructure, which stands in contrast to the LoRaWAN network paradigm.

There are three future directions for this work. First, studying allocation strategies DSME-LoRa networks may potentially facilitate real-time communication over LoRa. Second, optimizing its operation on battery-powered devices, aiming to achieve deterministic wireless communications at high energy efficiency. Third, the evaluation of SCHC compression over DSME-LoRa shall open new research possibilities.

7 References

[1] J. Alamos, P. Kietzmann, T. C. Schmidt, and M. Wählisch. DSME-LoRa: Seamless Long Range Communication Between Arbitrary Nodes in the Constrained IoT. *Transactions on Sensor Networks (TOSN)*, 18(4):1–43, November 2022.

[2] E. Baccelli, C. Gündogan, O. Hahm, P. Kietzmann, M. Lenders, H. Petersen, K. Schleiser, T. C. Schmidt, and M. Wählisch. RIOT: an Open Source Operating System for Low-end Embedded Devices in the IoT. *IEEE Internet of Things Journal*, 5(6):4428–4440, December 2018.

[3] F. Battaglia, M. Collotta, L. Leonardi, L. L. Bello, and G. Patti. Novel extensions to enhance scalability and reliability of the ieee 802.15.4-dsme protocol. *Electronics*, 9(1):1–16, 2020.

[4] H. E. Elbsir, M. Kassab, S. Bhiri, and M. H. Bedoui. Evaluation of LoRaWAN Class B efficiency for downlink traffic. In *2020 16th Intern. Conf. on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 105–110, October 2020. IEEE.

[5] J. Finnegan, S. Brown, and R. Farrell. Evaluating the Scalability of LoRaWAN Gateways for Class B Communication in ns-3. In *IEEE Conference on Standards for Communications and Networking (CSCN'18)*, pages 1–6, Piscataway, NJ, USA, 2018. IEEE.

[6] O. Gimenez and I. Petrov. Static Context Header Compression and Fragmentation (SCHC) over LoRaWAN. RFC 9011, IETF, 4/2021.

[7] C. Gomez and A. Minaburo. Transmission of SCHC-compressed packets over IEEE 802.15.4 networks. ID – wip 03, IETF, July 2022.

[8] C. Gomez, A. Minaburo, L. Toutain, D. Barthel, and J. Zuniga. Ipv6 over lpwans: Connecting low power wide area networks to the internet (of things). *IEEE Wireless Communications*, 27(1):206–213, 2020.

[9] J. Hui and P. Thubert. Compression Format for IPv6 Datagrams over IEEE 802.15.4-Based Networks. RFC 6282, IETF, September 2011.

[10] IEEE 802.15 Working Group. IEEE Standard for Low-Rate Wireless Networks. Technical Report IEEE Std 802.15.4™–2015 (Revision of IEEE Std 802.15.4-2011), IEEE, New York, NY, USA, 2016.

[11] LoRa Alliance – Technical Committee. Lorawan 1.1 specification. Technical report, LoRa Alliance, Oct. 2017.

[12] K. Mikhaylov, J. Petäjäjärvi, and A. Pouttu. Effect of Downlink Traffic on Performance of LoRaWAN LPWA Networks: Empirical Study. In *29th Intern. Symp. on Personal, Indoor and Mobile Radio Communications (PIMRC'18)*, Piscataway, NJ, USA, 2018. IEEE.

[13] A. Minaburo, L. Toutain, and R. Andreasen. Static Context Header Compression (SCHC) for the Constrained Application Protocol (CoAP). RFC 8824, IETF, June 2021.

[14] A. Minaburo et al.. SCHC: Generic Framework for Static Context Header Compression and Fragmentation. RFC 8724, IETF, 4/2020.

[15] G. Montenegro et al.. Transmission of IPv6 Packets over IEEE 802.15.4 Networks. RFC 4944, IETF, September 2007.

[16] B. Moons, A. Karaagac, J. Haxhibeqiri, E. D. Poorter, and J. Hoebeke. Using schc for an optimized protocol stack in multimodal lpwan solutions. In *2019 IEEE WF-IoT*, pages 430–435, 2019.

[17] R. Saroui, A. Guitton, O. Iova, and F. Valois. Uplink and downlink are not orthogonal in lorawan! In *2022 IEEE 96th Vehicular Technology Conference (VTC2022-Fall)*, pages 1–4, 2022.

[18] Z. Shelby, K. Hartke, and C. Bormann. The Constrained Application Protocol (CoAP). RFC 7252, IETF, June 2014.

[19] Y. Shiferaw, A. Arora, and F. Kuipers. LoRaWAN Class B Multicast Scalability. In *Proc. of 19th IFIP Networking Conference*, pages 609–613, Piscataway, NJ, USA, June 2020. IEEE Press.

[20] C. Vallati, S. Brienza, M. Palmieri, and G. Anastasi. Improving Network Formation in IEEE 802.15.4e DSME. *Computer Communications*, 114(C), December 2017.

[21] V. D. Vincenzo, M. Heusse, and B. Tourancheau. Improving Downlink Scalability in LoRaWAN. In *IEEE ICC'19*, 2019. IEEE.

[22] D. Zorbas et al.. TS-LoRa: Time-slotted LoRaWAN for the Industrial Internet of Things. *ComComm*, 153:1–10, 2020.